IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application No.:	10/663,640	Examiner:	Sajeda Muhebbullah
Atty. Docket No.:	SUN04-0196	Art Unit:	2174
Filed:	15 September 2003	Conf. No.:	1906
Appellant:	Kawahara et al.		

For: *Enhancements for Manipulating Two-Dimensional Windows within a Three-Dimensional Display Model*

APPEAL BRIEF

Sir:

Subsequent to the Notice of Appeal filed 27 August 2008, Appellant submits this Appeal Brief to appeal the rejection of claims 1, 6, 8-9, 13, 18, 20-21, 25, 30, 32-33, and 37-40 under 35 U.S.C. § 102 in a Final Office Action mailed 27 May 2008. This Appeal Brief demonstrates that such rejections cannot be sustained because each and every element as set forth in the claim is not found either expressly or inherently described in the cited prior art reference.

TABLE OF CONTENTS

THE REAL PARTY IN INTEREST

The real party in interest in this appeal is Sun Microsystems, Inc., the assignee of this application.

RELATED APPEALS AND INTERFERENCES

Appellant is not aware of any appeals or interferences that will affect directly, will be affected directly by, or will otherwise have bearing on the decision in this appeal.

For completeness, the Appellant notes the following applications which are related to this application:

1. Pending U.S patent app. no. 10/663,609, filed on 15 September 2003 (atty. docket no.: SUN04-0195).

STATUS OF CLAIMS

The status of the claims is as follows:

Claims pending:	1-3, 5-15, 17-27, and 29-40
Claims rejected:	1-3, 5-15, 17-27, and 29-40
Claims objected to:	None
Claims cancelled:	4, 16, and 28
Claims appealed:	1-3, 5-15, 17-27, and 29-40

STATUS OF AMENDMENTS

All amendments have been entered. A copy of the rejected claims is attached as appendix A.

SUMMARY OF THE CLAIMED SUBJECT MATTER

The claims in the instant application are directed towards a method, a computer readable storage medium, an apparatus, and a means for manipulating a window within a three-dimensional (3D) display model.

As described in the instant application,[1] many computer systems and other devices support window-based graphical user interfaces (GUIs) which were originally developed several decades ago. Due to limitations on processing power at that time, these GUIs were designed with simplified two-dimensional (2D) user interfaces. Based on a number of considerations (e.g., support for legacy systems, trained user base, etc.), manufacturers and vendors have continued to use 2D interfaces in modern computer systems. However, as newer generations of computers and devices with more processing power have been introduced, a number of 3D user interfaces have been developed. These 3D interfaces typically allow a user to navigate through and manipulate 3D objects. However, these 3D interfaces are mainly focused on exploiting newly developed 3D capabilities, while little attention has been given to supporting existing, legacy window-based 2D applications within these 3D user interfaces.

The claimed invention addresses this problem by enabling the manipulation of a 2D window in a 3D display model.

In summary, a system is described in paragraphs [0052]-[0060] of the instant application; operations on windows in a 3D display model and processes for performing these operations are described in paragraphs [0061]-[0076] of the instant application. The operations include:

(1) rotation of objects around a viewpoint (paragraph [0061]);
(2) rotating windows (paragraphs [0062]-[0065]);

---

[1] see instant application, paragraphs [0003]-[0006]

(3) minimizing windows (paragraphs [0066]-[0068]);
(4) tilting windows (paragraphs [0069]-[0070]);
(5) displaying information on the back of a window in (paragraph [0070]);
(6) using window controls on the side of a window (paragraph [0072]);
(7) minimizing top-level windows (paragraphs [0073]-[0074]); and
(8) throwing a window (paragraphs [0075]-[0076]).

Independent Claim 1: A Method for Manipulating a Window within a Three-Dimensional Display Model

The claimed method ("the method") enables the manipulation of a window within a three-dimensional (3D) display model. The method is described in paragraphs [0060]-[0076] of the instant application. A system for providing the 3D display model is described in paragraphs [0052]-[0060] of the instant application.

The method involves displaying a view into the 3D display model through a two-dimensional (2D) display. The 3D display model and the 2D display are described in paragraphs [0052]-[0060] of the instant application.

The method further involves receiving a command to manipulate the window within the 3D display model, wherein the window provides a 2D user interface for a 2D application. Receiving a command is described in paragraph [0064] and paragraphs [0057], [0061], [0066], [0069], [0071], [0072], [0073], and [0075] of the instant application.

In response to the command, the method involves manipulating the window within the 3D display model so that the manipulation is visible within the 2D display, wherein manipulating the window involves moving or rotating the window within the 3D display model around a viewpoint or around another point within the 3D display model. Manipulating the window within the 3D display model is described in paragraphs [0053]-[0054], [0057]-[0059] of the instant application. Rotating the window around a viewpoint or another point in the 3D display is described in paragraph [0064] of the instant application.

Dependent Claim 2: A Method for Manipulating a Window within a Three-Dimensional Display Model

Dependent claim 2 depends upon independent claim 1. In the claimed method, if the command moves the window in close proximity to an edge of the 2D display, the method further comprises tilting the window so that the window appears at an oblique angle in the 2D display. Tilting the window in response to moving the window in close proximity to an edge of the 2D display is described in paragraph [0069] of the instant application.

Dependent Claim 3: A Method for Manipulating a Window within a Three-Dimensional Display Model

Dependent claim 3 depends upon dependent claim 2. In the claimed method, wherein if the window is selected, the method further comprises untilting the window so that the window is parallel with the 2D display. Untilting the window is described in paragraph [0070] of the instant application.

Dependent Claim 5: A Method for Manipulating a Window within a Three-Dimensional Display Model

Dependent claim 5 depends upon dependent claim 38. In the claimed method, the information associated with the 2D application includes at least one of application version information; application settings; application parameters; application properties; and notes associated with a file or a web page that is displayed in the window. The information associated with the 2D application is described in paragraph [0071] of the instant application.

Dependent Claim 6: A Method for Manipulating a Window within a Three-Dimensional Display Model

Dependent claim 6 depends upon dependent claim 38. In the claimed method, the backside of the window includes the ability to accept user input, including change settings, parameters, properties and/or notes. The backside of the window is described in paragraph [0071] of the instant application.

Dependent Claim 7: A Method for Manipulating a Window within a Three-Dimensional Display Model

Dependent claim 7 depends upon independent claim 1. In the claimed method, if the command is to minimize the window, manipulating the window involves: (1) tilting the window so that a spine located on a side edge of the window is visible and the contents of the window remains visible, wherein the spine contains identification information for the window; and (2) moving the minimized window to an edge of the 2D display, wherein the operations of turning and moving the window are animated as a continuous motion. Minimizing the window is described in paragraph [0073] of the instant application.

Dependent Claim 8: A Method for Manipulating a Window within a Three-Dimensional Display Model

Dependent claim 8 depends upon independent claim 1. In the claimed method, the method further comprises receiving a predefined gesture through a pointing device. In response to the predefined gesture, the method involves minimizing a top-level window in the 2D display, whereby repeating the predefined gesture causes subsequent top-level windows to be minimized. Minimizing top-level windows is described in paragraph [0073] of the instant application.

Dependent Claim 9: A Method for Manipulating a Window within a Three-Dimensional Display Model

Dependent claim 9 depends upon dependent claim 8. In the claimed method, upon receiving a window restoration command, the method further comprises restoring minimized windows to their expanded state. Restoring windows is described in paragraph [0074] of the instant application.

Dependent Claim 10: Method for Manipulating a Window within a Three-Dimensional Display Model

Dependent claim 10 depends upon independent claim 1. In the claimed method, if the command is entered through a pointing device and the command throws the window by moving the window quickly and releasing it, the method further comprises throwing the window by moving the window in a continuous animated motion. Throwing the window is described in paragraphs [0075]-[0076] of the instant application.

Dependent Claim 11: A Method for Manipulating a Window within a Three-Dimensional Display Model

Dependent claim 11 depends upon dependent claim 10. In the claimed method, throwing the window involves at least one of: locating the window farther from the viewpoint; scaling down the size of the window; iconizing the window; and deleting the window. Throwing the window is described in paragraphs [0075]-[0076] of the instant application.

Dependent Claim 12: A Method for Manipulating a Window within a Three-Dimensional Display Model

Dependent claim 12 depends upon independent claim 1. In the claimed method, receiving the command involves: rotating the window so that window controls on the edge of the window become visible in response to a cursor moving close to an edge of a window; receiving the command through a window control; and rotating the window back to its original orientation. Rotating the window to show window controls and receiving command through the window controls is described in paragraphs [0062]-[0065] and [0072] of the instant application.

Dependent Claim 38: A Method for Manipulating a Window within a Three-Dimensional Display Model

Dependent claim 38 depends upon independent claim 1. In the claimed method, if the command rotates the window so that the backside of the window is visible, the method further comprises displaying information associated with the 2D application on the backside of the window. Displaying information on the backside of the window is described in paragraph [0071] of the instant application.

Independent Claim 13 and Dependent Claims 14-15, 17-24, and 39: A Computer-Readable Storage Medium Storing Instructions that when Executed by a Computer cause the Computer to Perform a Method for Manipulating a Window within a Three-Dimensional Display Model

Much of the subject matter in independent claim 1 and dependent claims 2-3, 5-12, and 38 also appears in independent claim 13 and dependent claims 14-15, 17-24, and 39, respectively, as applied to a computer-readable storage medium. Aside from the computer readable storage medium, which is described in paragraph [0051] of the instant application, the remaining subject matter of

claims 1, 2-3, 5-12, and 38, as summarized above, is sufficient to establish patentability. Appellant therefore does not repeat the above description.

Independent Claim 25 and Dependent Claims 26-27, 29-36, and 40: An Apparatus for Manipulating a Window within a Three-Dimensional Display Model

Much of the subject matter in independent claim 1 and dependent claims 2-3, 5-12, and 38 also appears in independent claim 25 and dependent claims 26-27, 29-36, and 40, respectively, as applied to an apparatus. Aside from the apparatus, which is described in paragraphs [0052]-[0060] of the instant application, the remaining subject matter of claims 1, 2-3, 5-12, and 38, as summarized above, is sufficient to establish patentability. Appellant therefore does not repeat the above description.

Independent Claim 37: A Means for Manipulating a Window within a Three-Dimensional Display Model

The independent means plus function claim includes a two-dimensional (2D) display means for displaying a view into the 3D display model. The 3D display model and the 2D display means are described in paragraphs [0052]-[0060] of the instant application.

The means plus function claim further includes a window manipulation means configured to receive a command to manipulate the window within the 3D display model, wherein the window provides a 2D user interface for a 2D application. Receiving a command is described in paragraph [0064] and paragraphs [0057], [0061], [0066], [0069], [0071], [0072], [0073], and [0075] of the instant application.

In response to the command, the window manipulation means manipulates the window within the 3D display model so that the manipulation is visible within

the 2D display, wherein manipulating the window involves moving or rotating the window within the 3D display model around a viewpoint or around another point within the 3D display model. Manipulating the window within the 3D display model is described in paragraphs [0053]-[0054], [0057]-[0059] of the instant application. Rotating the window around a viewpoint or another point in the 3D display is described in paragraph [0064] of the instant application.

GROUNDS OF REJECTION PRESENTED FOR REVIEW

In the final Office Action mailed 25 April 2008, the Examiner reviewed claims 1-3, 5-15, 17-27, and 29-40. The Examiner rejected claims 1, 6, 8-9, 13, 18, 20-21, 25, 30, 32-33, and 37-40 under 35 U.S.C. § 102(e) as being anticipated by Miller (U.S. patent no. 6,597,358, hereinafter "Miller").

For the purposes of this appeal, and without admission as to the appropriateness of the other grounds raised by the Examiner, Appellants address the Examiner's reliance on Miller for disclosing moving or rotating a window within a 3D display model **around a viewpoint or around another point within the 3D display model**. More specifically, Appellants demonstrate that in embodiments of the present invention, the windows can rotate around a point within the 3D display model or around *a point outside of the 3D display model*, such as a point of view of a user, etc.

ARGUMENTS

Rejections under 35 U.S.C. § 102(e)

In response to the rejection under 35 U.S.C. § 102(e) in the final Office Action mailed 27 May 2008 (hereinafter "0527 OA"), Appellant submits that the rejections cannot be sustained because:

When establishing a prima facie case when rejecting claims under 35 U.S.C. § 102(e), the Examiner's cited prior art must contain every element of the claimed subject matter:

> A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.[2]

Examiner has failed to establish prima facie anticipation because there is at least one fundamental distinction between the cited Miller prior art and the claimed invention. Specifically, there is nothing in Miller, either expressly or inherently, that discloses moving or rotating a window within a 3D display model around a viewpoint or around another point within the 3D display model.

Overview of Miller

In the interest of clarifying the arguments against the rejection of the claimed invention using Miller, we provide an overview of the system disclosed in Miller. In addition to providing the overview, we briefly identify the limitations of the claimed invention that are missing from the Miller system.

---

[2] see Manual of Patent Examining Procedure (MPEP) § 2131

Generally, the Miller system is dedicated to organizing 2D and/or 3D computer applications for display in 3D viewing perspective in a 2D display environment.[3] In the Miller system, a 2D or 3D application generates an application window that is to be displayed to a user. The Miller system then renders the application window (i.e., creates a texture) so that the application window can be placed on one face of a 3D cube in a 2D display environment for viewing by a user.[4] Rendering application windows is described by Miller as follows:

> In conventional 3D graphics, "texture" (i.e., images) from bitmaps in the texture memory 260 are mapped onto three-dimensional geometries, known as "primitives" generated by the graphics rendering unit 250. The process of applying this texture to such primitives, known as "rendering", is well established in the art;[5] and
>
> The graphics rendering unit 250 generates a series of primitives for applying the bitmaps 310 of the two and three-dimensional computer applications that are to be mapped thereunto. In one embodiment, the primitives generated by the graphics rendering unit 250 are a series of "planes" as shown by the 3D application windows 420 in FIG. 4.[6]

Miller discloses a number of techniques for interacting with the application windows that are located on the 3D cube's faces. One of Miller's interactions involves rotating the 3D cube to enable the user to view and interact with the application windows displayed on different faces of the 3D cube. For example, in the Miller system, a user can click on the 3D cube to cause the 3D cube to rotate so that the contents of a window that were in the top face or a side

---

[3] see Miller, Abstract

[4] see Miller, FIG. 6 (specifically, application windows *A*, *B* and *E* in FIG. 6)

[5] see Miller, col. 5, lines 23-27

[6] see Miller, col. 5, lines 38-43

face of the 3D cube are located on the front face of the 3D cube. The following summarizes Miller's description of rotating the 3D cube:

> The contents of the application windows could be mapped onto the surfaces of a three-dimensional cube by the graphics rendering unit. The user could view all the surfaces of the cube, and, thus, all of the computer applications mapped thereon, by **rotating the cube** in a predetermined direction via the user-input device. This may be accomplished by receiving a first predetermined input at the user-input device to cause the cube to **rotate horizontally**. Or, alternatively, receiving a second predetermined input that may cause the cube to **rotate vertically**. Such rotation could be accomplished by animating the movement of the cube either horizontally or vertically, which could be achieved using techniques that are well established in the art of computer graphics.[7]

Although describing **rotating the 3D cube in place** and moving the cube by "dragging" the cube within the display window,[8] Miller nowhere describes moving or rotating a window within a 3D display model around a viewpoint or around another point within the 3D display model. As is well-known in the art (and in engineering generally), rotating an object (i.e., the 3D cube) involves rotating the object about a central axis of the object.

Rotating an object around a center axis of the object is not equivalent to rotating a window within a 3D display model **around a viewpoint** (see, e.g., viewpoint 106 in FIG. 1 of the instant application). Moreover, rotating an object around a center axis of the object is not equivalent to rotating a window around **another point** within the 3D display model. As an example of the distinction, consider the motions of the planets in the solar system. The planets rotate around a *center axis* (like the 3D cube in Miller). However, the planets also rotate around *other points*: the Sun, the galactic center, and myriad other points (like the instant application's rotation around a viewpoint or another point in the 3D

---

[7] see Miller, col. 6, lines 37-52

[8] see Miller col. 6, lines 53-57

model). Thus, in Miller's rotation around a center axis, the 3D windows *rotate in place*, whereas in rotating around the viewpoint or another point in the 3D model, the instant application's windows *move* in space around that point.

Rejections of Independent Claims 1, 13, 25, and 37

The Examiner rejected independent claims 1, 13, 25, and 37 under 35 U.S.C. § 102(e) as being anticipated by Miller. Appellant respectfully disagrees with the rejection. The rejection of independent claims 1, 13, 25, and 37 is improper because there is at least one fundamental distinction between the cited Miller prior art and the claimed invention. Specifically, there is nothing in Miller, either expressly or inherently, that discloses moving or rotating a window within a 3D display model around a viewpoint or around another point within the 3D display model. Appellant addresses this point in the following section.

Note that because claim 1, which is directed to a method, claim 13, which is directed to a computer-readable storage medium, claim 25, which is directed to an apparatus, and claim 37, which is directed to a means for manipulating a window include a similarly rejected limitation, Appellant addresses the rejections for all of the claims in the following section.

1. Miller cannot Anticipate the Claimed Invention because Each and Every Element as Set Forth in the Claim is not Found, either Expressly or Inherently Described, in Miller

The Examiner has failed to establish prima facie anticipation because the Examiner has not shown that Miller discloses each and every element of the claimed invention. Specifically, the Examiner has not shown that Miller discloses moving or rotating a window within a 3D display model around a viewpoint or around another point within the 3D display model.

Miller Nowhere Discloses a Moving or Rotating a Window within a 3D Display Model around a Viewpoint or around another Point within the 3D Display Model

In the 0527 OA, the Examiner rejected the independent claims of the instant application, arguing that the limitation:

> "in response to the command, manipulating the window within the 3D display model so that the manipulation is visible within the 2D display; and wherein manipulating the window involves moving or rotating the window within the 3D display model around a viewpoint or around another point within the 3D display model"

of the instant application was anticipated by Miller.[9] More specifically, the Examiner argues that Miller's description of a "ray-picking process" is sufficient to anticipate the limitation. Appellant respectfully disagrees with the rejection.

As described in the overview section above, Miller is expressly limited to rotating the 3D cube in-place or moving the 3D cube by "dragging" the 3D cube within the display environment.[10] As is well-known in the art (and in engineering generally), rotating an object involves rotating the object about a central axis of the object. Thus, Miller nowhere discloses moving or rotating a window within a 3D display model **around a viewpoint** or around **another point within the 3D display model**.

In the section of Miller cited by the Examiner in support of the rejection of the independent claims,[11] Miller describes only a commonly-known "ray-picking process" that is used to determine a location in the simulated 3D environment where a user has performed an action. The following summarizes Miller's description of the ray-picking process:

---

[9] see 0527 OA, page 3, first paragraph

[10] see Miller, col. 6, lines 37-57

[11] see 0527 OA, page 3, first paragraph, citing Miller, col. 6, lines 32-52

> The user interacts with the "3D" window utilizing the user-input device (e.g., a pointing device such as a mouse). As practiced in the art of 3D graphics programming, the present invention can **process the 2D events generated by the user-input device and project their 2D position into the 3D graphical space to detect when the pointer (from the user-input device) is over the 3D object textured with the application window.** This "*ray-picking*" process can identify not only the 3D object, but can also determine the pointer's position in the coordinates of the texture of the 3D geometry to which it is applied. Since this 2D texture coordinate corresponds directly to the 2D coordinates of the application window, the present invention can easily synthesize another pointer event, this time in the coordinate space of the application window, and send that event to that window, as is well known in the art of 2D graphical user interface programming. **When the application window receives this event, it will typically behave as if the pointer event occurred over the actual window, when in fact it originally occurred in the "3D" application window.**[12]

Although describing a well-known technique for interacting with an object in a simulated 3D environment on a 2D display, the cited section of Miller nowhere discloses moving or rotating a window within a 3D display model around a viewpoint or around another point within the 3D display model. (Note that the instant application also describes using the well-known technique of ray tracing "for mapping objects from 3D display model into corresponding locations in 2D display."[13])

In contrast to Miller, the claimed invention allows a user to move or rotate objects within the 3D display model **around a viewpoint or another point within the 3D display model**.[14] In other words, the claimed invention enables a user to move a window from one location to another within the 3D display model or to rotate a window within the 3D display model around a predetermined point

---

[12] see Miller, col. 6, lines 13-33

[13] see instant application, paragraph [0054]

[14] see instant application, par. [0052] and [0061]

in the 3D display model or around a predetermined point *outside of the 3D display model*. For example, FIG. 1 of the instant application clearly shows **viewpoint 106**, which is outside the 3D display model 102. In the claimed invention, the rotational motion makes it easier for a user to identify window boundaries and also gives the user a feeling of depth and space.[15]

Miller's ray-picking is not equivalent to moving or rotating a window within the 3D display model around a viewpoint or another point. In addition, Miller's in-place rotation of a 3D cube is not equivalent to moving or rotating a window within the 3D display model around a viewpoint or another point.

[15] see instant application, par. [0061]

Conclusion

In summary, Appellant has demonstrated that the rejections under 35 U.S.C. § 102(e) cannot be sustained because Miller does not disclose moving or rotating a window within a 3D display model around a viewpoint or around another point within the 3D display model.

In view of the foregoing, Appellant respectfully requests the reversal of all of the rejections in the Final Office Action mailed 27 May 2008. Appellant further requests allowance of all the claims in the instant application.

Respectfully submitted,

By: /Anthony P. Jones/
Anthony P. Jones
Registration No. 59,521

Date: 21 November 2008

Anthony Jones
Park, Vaughan & Fleming LLP
2820 Fifth Street
Davis, CA 95618-7759
Tel: (530) 759-1666
Fax: (530) 759-1665
Email: tony@parklegal.com

APPENDICES

Appendix A: Claims Appendix

1. (Previously Presented) A method for manipulating a window within a three-dimensional (3D) display model, comprising:

displaying a view into the 3D display model through a two-dimensional (2D) display;

receiving a command to manipulate the window within the 3D display model, wherein the window provides a 2D user interface for a 2D application;

in response to the command, manipulating the window within the 3D display model so that the manipulation is visible within the 2D display; and

wherein manipulating the window involves moving or rotating the window within the 3D display model around a viewpoint or around another point within the 3D display model.

2. (Original) The method of claim 1, wherein if the command moves the window in close proximity to an edge of the 2D display, the method further comprises tilting the window so that the window appears at an oblique angle in the 2D display, whereby the contents of the window remain visible, while the window occupies less space in the 2D display and is less likely to overlap other windows.

3. (Original) The method of claim 2, wherein if the window is selected, the method further comprises untilting the window so that the window is parallel with the 2D display.

4. (Canceled).

5. (Previously Presented) The method of claim 38, wherein the information associated with the 2D application includes at least one of:

application version information;

application settings;

application parameters;

application properties; and

notes associated with a file or a web page that is displayed in the window.

6. (Previously Presented) The method of claim 38, wherein the backside of the window includes the ability to accept user input, including change settings, parameters, properties and/or notes.

7. (Original) The method of claim 1, wherein if the command is to minimize the window, manipulating the window involves:

tilting the window so that a spine located on a side edge of the window is visible and the contents of the window remains visible, wherein the spine contains identification information for the window; and

moving the minimized window to an edge of the 2D display;

wherein the operations of turning and moving the window are animated as a continuous motion.

8. (Original) The method of claim 1, further comprising:

receiving a predefined gesture through a pointing device, and

in response to the predefined gesture, minimizing a top-level window in the 2D display, whereby repeating the predefined gesture causes subsequent top-level windows to be minimized.

9. (Original) The method of claim 8, wherein upon receiving a window restoration command, the method further comprises restoring minimized windows to their expanded state.

10. (Original) The method of claim 1, wherein if the command is entered through a pointing device and the command throws the window by moving the window quickly and releasing it, the method further comprises throwing the window by moving the window in a continuous animated motion.

11. (Previously Presented) The method of claim 10, wherein throwing the window involves at least one of:

locating the window farther from the viewpoint;

scaling down the size of the window;

iconizing the window; and

deleting the window.

12. (Original) The method of claim 1, wherein receiving the command involves:

rotating the window so that window controls on the edge of the window become visible in response to a cursor moving close to an edge of a window;

receiving the command through a window control; and

rotating the window back to its original orientation.

13. (Previously Presented) A computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform a method for manipulating a window within a three-dimensional (3D) display model, the method comprising:

displaying a view into the 3D display model through a two-dimensional (2D) display;

receiving a command to manipulate the window within the 3D display model, wherein the window provides a 2D user interface for a 2D application;

in response to the command, manipulating the window within the 3D display model so that the manipulation is visible within the 2D display; and

wherein manipulating the window involves moving or rotating the window within the 3D display model around a viewpoint or around another point within the 3D display model.

14. (Original) The computer-readable storage medium of claim 13, wherein if the command moves the window in close proximity to an edge of the 2D display, the method further comprises tilting the window so that the window appears at an oblique angle in the 2D display, whereby the contents of the window remain visible, while the window occupies less space in the 2D display and is less likely to overlap other windows.

15. (Original) The computer-readable storage medium of claim 14, wherein if the window is selected, the method further comprises untilting the window so that the window is parallel with the 2D display.

16 (Canceled).

17. (Previously Presented) The computer-readable storage medium of claim 39, wherein the information associated with the 2D application includes at least one of:

application version information;

application settings;

application parameters;

application properties; and

notes associated with a file or a web page that is displayed in the window.

18. (Previously Presented) The computer-readable storage medium of claim 39, wherein the backside of the window includes the ability to accept user input, including change settings, parameters, properties and/or notes.

19. (Original) The computer-readable storage medium of claim 13, wherein if the command is to minimize the window, manipulating the window involves:

tilting the window so that a spine located on a side edge of the window is visible and the contents of the window remains visible, wherein the spine contains identification information for the window; and

moving the minimized window to an edge of the 2D display;

wherein the operations of turning and moving the window are animated as a continuous motion.

20. (Original) The computer-readable storage medium of claim 13, wherein the method further comprises:

receiving a predefined gesture through a pointing device, and

in response to the predefined gesture, minimizing a top-level window in the 2D display, whereby repeating the predefined gesture causes subsequent top-level windows to be minimized.

21. (Original) The computer-readable storage medium of claim 20, wherein upon receiving a window restoration command, the method further comprises restoring minimized windows to their expanded state.

22. (Original) The computer-readable storage medium of claim 13, wherein if the command is entered through a pointing device and the command throws the window by moving the window quickly and releasing it, the method further comprises throwing the window by moving the window in a continuous animated motion.

23. (Previously Presented) The computer-readable storage medium of claim 22, wherein throwing the window involves at least one of:

locating the window farther from the viewpoint;

scaling down the size of the window;

iconizing the window; and

deleting the window.

24. (Original) The computer-readable storage medium of claim 13, wherein receiving the command involves:

rotating the window so that window controls on the edge of the window become visible in response to a cursor moving close to an edge of a window;

receiving the command through a window control; and

rotating the window back to its original orientation.

25. (Previously Presented) An apparatus that manipulates a window within a three-dimensional (3D) display model, comprising:

a two-dimensional (2D) display configured to display a view into the 3D display model;

a window manipulation mechanism configured to receive a command to manipulate the window within the 3D display model, wherein the window provides a 2D user interface for a 2D application;

wherein in response to the command, the window manipulation mechanism is configured to manipulate the window within the 3D display model so that the manipulation is visible within the 2D display; and

wherein manipulating the window involves moving or rotating the window within the 3D display model around a viewpoint or around another point within the 3D display model.

26. (Original) The apparatus of claim 25, wherein if the command moves the window in close proximity to an edge of the 2D display, the window manipulation mechanism is configured to tilt the window so that the window appears at an oblique angle in the 2D display, whereby the contents of the window remain visible, while the window occupies less space in the 2D display and is less likely to overlap other windows.

27. (Original) The apparatus of claim 26, wherein if the window is selected, the window manipulation mechanism is configured to untilt the window so that the window is parallel with the 2D display.

28. (Canceled)

29. (Previously Presented) The apparatus of claim 40, wherein the information associated with the 2D application includes at least one of:

application version information;

application settings;

application parameters;

application properties; and

notes associated with a file or a web page that is displayed in the window.

30. (Previously Presented) The apparatus of claim 40, wherein the backside of the window includes the ability to accept user input, including change settings, parameters, properties and/or notes.

31. (Original) The apparatus of claim 25, wherein if the command is to minimize the window, the window manipulation mechanism is configured to:

tilt the window so that a spine located on a side edge of the window is visible and the contents of the window remains visible, wherein the spine contains identification information for the window; and to

move the minimized window to an edge of the 2D display;

wherein the operations of turning and moving the window are animated as a continuous motion.

32. (Original) The apparatus of claim 25, wherein the window manipulation mechanism is additionally configured to:

receive a predefined gesture through a pointing device, and

in response to the predefined gesture, to minimize a top-level window in the 2D display, whereby repeating the predefined gesture causes subsequent top-level windows to be minimized.

33. (Original) The apparatus of claim 32, wherein upon receiving a window restoration command, the window manipulation mechanism is configured to restore minimized windows to their expanded state.

34. (Original) The apparatus of claim 25, wherein if the command is entered through a pointing device and the command throws the window by moving the window quickly and releasing it, the window manipulation

mechanism is configured to throw the window by moving the window in a continuous animated motion.

35. (Previously Presented) The apparatus of claim 34, wherein throwing the window involves at least one of:

locating the window farther from the viewpoint;

scaling down the size of the window;

iconizing the window; and

deleting the window.

36. (Original) The apparatus of claim 25, wherein while receiving the command, the window manipulation mechanism is configured to:

rotate the window so that window controls on the edge of the window become visible in response to a cursor moving close to an edge of a window;

receive the command through a window control; and to

rotate the window back to its original orientation.

37. (Previously Presented) A means for manipulating a window within a three-dimensional (3D) display model, comprising:

a two-dimensional (2D) display means for displaying a view into the 3D display model;

a window manipulation means configured to receive a command to manipulate the window within the 3D display model, wherein the window provides a 2D user interface for a 2D application;

wherein in response to the command, the window manipulation means manipulates the window within the 3D display model so that the manipulation is visible within the 2D display; and

wherein manipulating the window involves moving or rotating the window within the 3D display model around a viewpoint or around another point within the 3D display model.

38. (Previously Presented) The method of claim 1, wherein if the command rotates the window so that the backside of the window is visible, the method further comprises displaying information associated with the 2D application on the backside of the window.

39. (Previously Presented) The computer-readable storage medium of claim 13, wherein if the command rotates the window so that the backside of the window is visible, the method further comprises displaying information associated with the 2D application on the backside of the window.

40. (Previously Presented) The apparatus of claim 25, wherein if the command rotates the window so that the backside of the window is visible, the method further comprises displaying information associated with the 2D application on the backside of the window.

Appendix B: Evidence

For this appeal, Appellants do not rely on any evidence submitted pursuant to §§ 1.130, 1.131, or 1.132, or other evidence entered by the Examiner.

Appendix C: Related Proceedings

Appellants are aware of no related proceedings.